IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

ATTORNEY DOCKET NO. AT9-98-781

In re Application of:	§	
	§	
ARIMILLI	§	
	§	Examiner: **TZENG, F.**
Serial No. **09/340,074**	§	
	§	
Filed: **25 JUNE 1999**	§	Art Unit: **2186**
	§	
For: **LAYERED LOCAL CACHE WITH**	§	
LOWER LEVEL CACHE OPTIMIZING	§	
ALLOCATION MECHANISM	§	

APPEAL BRIEF

Hon. Commissioner of Patents
and Trademarks
Washington, D.C. 20231

Sir:

This Brief is submitted in triplicate in support of the Appeal in the above-identified application.

REAL PARTY IN INTEREST

International Business Machines Corporation, the assignee of record, is the real party in interest in the subject Appeal.

RELATED APPEALS AND INTERFERENCES

The only pending appeal known to Appellant, Appellant's legal representative, or assignee that will directly affect or be directly affected by or have a bearing on the Board's decision in the present Appeal is the appeal of U.S. Patent Application Serial No. 09/340,075. However, as noted in the cross-reference section of the present specification, the present application is related to a number of other applications, many of which are under final rejection and are therefore likely to be the subject of future appeals.

STATUS OF THE CLAIMS

Claims 1-7, 10-18 and 21-27 are pending, and Claims 8-9 and 19-20 have been canceled. All pending claims stand finally rejected by the Examiner as noted in the Final Rejection dated June 5, 2001, and labeled Paper No. 4. The rejection of each pending claim is appealed.

STATUS OF AMENDMENTS

No amendment was proposed or entered subsequent to the Final Rejection.

SUMMARY OF THE INVENTION

The present invention is directed to a data processing system having an improved multi-level cache hierarchy and an improved method of operating the multi-level cache hierarchy of a data processing system. In particular, the present invention provides an improved method and system by which a victim cache block in a lower level cache is selected for replacement based, at least in part, on cache hits in an upper level cache.

As depicted in Figures 2 and 3 and as described in detail at page 13, line 1 through page 18, line 18 of the present specification, a suitable data processing system 120 in which the present

invention may be practiced includes a central processing unit (CPU) 150 having a multi-level cache hierarchy. As described at page 18, line 23 *et seq*. and as illustrated in Figure 4, the cache hierarchy includes at least an upper level cache 200 and a lower level cache 202. Caches 200 and 202 are both coupled to a request bus 212 so that each of caches 200 and 202 receives each data access request of the load-store unit (LSU) 204 of CPU 150 (page 19, lines 11-13). L1 data directory 208 provides a flag to L2 controller 214 for each request to indicate whether or not the request hit or missed L1 cache 200.

As depicted in Figure 4 and as described at page 21, line 26 through page 22, line 7, L2 controller 214 maintains an L2 least recently used (LRU) array 232 that indicates which entry within L2 entry array 218 should be selected by victim select logic 234 for replacement if a processor request misses both L1 cache 200 and L2 cache 202. Prior art L2 LRU arrays are only updated based upon L1 cache <u>misses</u> because conventional L2 caches only receive accesses that miss the higher level L1 cache and do not receive any accesses that hit the L1 cache. Unlike these prior art L2 LRU arrays, L2 LRU 232 is updated by L2 controller 214 in response to <u>both</u> accesses that <u>hit</u> L1 cache 200 and accesses that <u>miss</u> L1 cache 200. Thus, L2 LRU 232 is a hybrid array that "includes information based on not only L1 misses, but further <u>on L1 hits</u>" (page 22, lines 4-5, emphasis supplied).

In this manner, cache lines that are frequently accessed by a processor and therefore result in L1 cache hits are more likely to be maintained in the L2 cache rather than evicted in response to an L2 cache miss. As a result, average data access latency is improved. The present invention is particularly advantageous when applied to inclusive cache hierarchies since the eviction of a cache line from the L2 cache of an inclusive cache hierarchy would also necessitate the eviction of a corresponding cache line from the L1 cache.

ISSUE

Is the Examiner's rejection of Claims 1-7, 10-18 and 21-27 under 35 U.S.C. § 102(b) as anticipated by U.S. Patent No. 5,737,751 to *Patel et al.* (*Patel*) well-founded?

PAGE 3
DOCKET NO. AT9-98-781

GROUPING OF THE CLAIMS

For purposes of this Appeal, Claims 1-7, 10-18 and 21-27 stand or fall together as a single group.

ARGUMENT

In paragraph 6 of the Final Rejection, Claims 1-7, 10-18 and 21-27 are rejected under 35 U.S.C. § 102(b) as anticipated by U.S. Patent No. 5,737,751 to *Patel et al.* (*Patel*). That rejection is also not well founded and should be reversed.

Applicant believes that *Patel* does not render the present claims unpatentable under 35 U.S.C. § 102 or § 103 because *Patel* does not identically disclose each feature of exemplary Claim 1. Exemplary Claim 1 recites a "method of operating a multi-level cache of a computer system," which following a miss in both the upper and lower level caches, "select[s] a victim cache block in the lower level cache for receiving the requested value based at least in part on cache <u>hits</u> in the upper level cache" (emphasis supplied). Thus, according to the present invention, a victim entry in a lower level (e.g., L2) cache is selected for replacement at least in part utilizing upper level (e.g., L1) cache <u>hit</u> information. As noted above, this recitation stands in direct contrast to conventional multilevel cache hierarchies in which only L1 <u>misses</u> (<u>not</u> hits) are made visible to lower level caches.

In paragraphs 4 and 6 of the present Office Action, the Examiner cites col. 3, lines 5-10 of *Patel* as teaching "selecting a victim cache block in the lower level cache for receiving the requested value based at least in part on cache hits in the upper level cache." In its entirety, the cited passage discloses:

> FIGS. 1A-1D illustrate the general cache line flow for cache requests which miss in the L1 and L2 caches. FIG. 1A shows the cache line flow when the tag entry look-up for both the L1 and L2 misses and no replacement copybacks are required for L1 or L2. As can be seen, the cache line is sent to both L1 and L2, but is queued in the L2 reload queue for the transfer to L2.
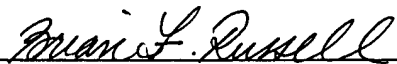
Thus, as correctly noted by the Examiner, this passage and Figure 1A of *Patel* teach that when a request of *Patel's* processor 12 misses both L1 cache 14 and L2 cache 20, the requested data are loaded directly into L1 cache 14, but must be queued in an L2 reload queue within processor 12 prior to transfer to L2 cache 20 via local bus 17.

Although the linefill operation disclosed by *Patel* may require selection of a victim cache block in the L2 cache, *Patel* clearly fails to mention selection of the victim cache block and certainly does not teach, suggest or motivate "selecting a victim cache block in the lower level cache ... based at least in part on cache hits in the upper level cache" as claimed. The method of victim cache block selection recited in exemplary Claim 1 is also not inherent in *Patel* because the particular method of victim cache block selection recited in the present claims is not required by *Patel*, as evidenced by the fact that an alternative method of victim selection (viz. one based solely on L1 cache misses) is commonly employed in the art.

Because *Patel* does not disclose the victim cache block selection recited in exemplary Claim 1 explicitly, inherently, or by suggestion, *Patel* fails to render the present claims unpatentable under 35 U.S.C. § 102 or § 103. Appellants therefore respectfully request that the Board reverse all rejections set forth in the Final Rejection.

Please charge Deposit Account No. **09-0447** in the amount of $320.00 for submission of a Brief in Support of Appeal. No additional fee is believed to be required; however, in the event an additional fee is required please charge that fee to Deposit Account No. **09-0447**. No extension of time is believed to be required; however, in the event an extension of time is required, please consider that extension requested and please charge any associated fee therefore to the above-identified Deposit Account No. **09-0447**.

Respectfully submitted,

Brian F. Russell
Registration No. 40,796
BRACEWELL & PATTERSON, L.L.P.
Suite 350 Lakewood on the Park
7600B N. Capital of Texas Hwy.
Austin, Texas 78731
(512) 343-6116

ATTORNEY FOR APPELLANTS

APPENDIX

1   1.   A method of operating a multi-level cache of a computer system, comprising the steps of:

2   monitoring cache activity of an upper level cache and a lower level cache both associated

3   with a processor of the computer system, said monitoring including monitoring cache hits in the

4   upper level cache;

5   issuing a request from the processor to load a value, wherein the request misses the upper

6   level cache and the lower level cache; and

7   selecting a victim cache block in the lower level cache for receiving the requested value

8   based at least in part on cache hits in the upper level cache.

1   2.   The method of Claim 1 wherein the victim cache block is further selected based in part on

2   the cache activity of the lower level cache.

1   3.   The method of Claim 1 wherein said selecting step takes place out of a critical path of

2   execution of a core of the processor.

1   4.   The method of Claim 1 wherein said issuing step issues a request to load operand data.

1   5.   The method of Claim 1 wherein said selecting step includes the step of identifying a less

2   recently used cache block in the lower level cache.

1   6.   The method of Claim 1 further comprising the steps of:

2   returning the requested value to the processor;

3   determining that it would be efficient to currently load into the upper level cache a cache line

4   which includes the requested value; and

·5   in response to said determining step, loading the cache line into the upper level cache.

1   7.   The method of Claim 1 wherein:

2   said monitoring step monitors cache misses of the upper level and lower level caches; and

3   said selecting step selects the victim cache block based at least in part on the cache misses

4   of the lower level cache.

     8. (canceled)

     9. (canceled)

1   10.   The method of Claim 1 further comprising the step of selecting a victim cache block in the

2   upper level cache for receiving the requested value based at least in part on the cache activity of the

3   lower level cache.

1   11.   A computer system comprising:

2   a system memory device;

3   means for processing program instructions;

4   means, connected to said processing means, for caching values stored in said system memory

5   device, said caching means having at least an upper level cache and a lower level cache both

6   associated with said processing means;

7   means for monitoring cache activity of said upper level cache and said lower level cache

8   including cache hits in the upper level cache; and

9   means for selecting a victim cache block in said lower level cache for receiving a value

10   specified in a load request issued by said processing means, wherein the load request missed said

11   upper level cache and said lower level cache, based at least in part on cache hits in said upper level

12   cache.

1   12.   The computer system of Claim 11 wherein said selecting means is located out of a critical

2   path of execution of a core of said processing means.

1   13.   The computer system of Claim 11 wherein said upper level cache is an operand data cache.

1   14.   The computer system of Claim 11 wherein said selecting means identifies a less recently used

2   cache block in said upper level cache.

1   15.   The computer system of Claim 11 wherein:

2   said upper level cache is an L1 cache; and

3   said lower level cache is an L2 cache.

1   16.   The computer system of Claim 11 wherein said upper level cache is a store-through cache.

1   17.   The computer system of Claim 11 further comprising:

2   means for returning the requested value to said processing means in response to the load

3   request missing said upper level cache; and

4   means for loading a cache line which includes the requested value into said upper level cache

5   in response to a determination that it would be efficient to currently load the cache line into said

6   upper level cache.

1   18.   The computer system of Claim 11 wherein:

2   said monitoring means monitors cache misses of said lower level cache; and

3   said selecting means selects said victim cache block based at least in part on the cache misses

4   of said lower level cache.

19. (canceled)

20. (canceled)

1   21.   The computer system of Claim 11 further comprising means for selecting a victim cache

2   block in said upper level cache for receiving the requested value based at least in part on the cache

3   activity of said lower level cache.

1   22.   A processing unit, comprising:

2   at least one instruction execution unit;

3   at least an upper level cache and a lower level cache;

4   a cache controller that, responsive to receipt of a load request issued by said at least one

5   execution unit that missed said upper level cache and said lower level cache, selects a victim cache

6   block in said lower level cache for receiving a value specified in the load request, wherein the

7   selection is based at least in part on cache hits in said upper level cache.

1   23.   The processing unit of Claim 22, wherein said upper level cache is an operand data cache.

1   24.   The processing unit of Claim 22, wherein said upper level cache is a store-through cache.

1   25.   The processing unit of Claim 22, further comprising:

2   means for loading a cache line including the requested value into said upper level cache in

3   response to a determination that it would be efficient to load the cache line into said upper level

4   cache.

1   26.   The processing unit of Claim 22, wherein said cache controller selects said victim cache

2   block based at least in part on the cache misses of said lower level cache.

1   27.   The processing unit of Claim 22, and further comprising means for selecting a victim cache

2   block in said upper level cache for receiving the requested value based at least in part on the cache

3   activity of said lower level cache.